2.8 Exercises

Derived Types:

1. Run the program `vehicle.f90`. What difference do you notice in the output of the two `WRITE` statements?

2. Run the program `circle1.f90`. Create a new derived type for a rectangle and assign and write out the corners of the rectangle. (`rectdef.f90`)

3. Create a file `circle.dat` which contains the components of the centre and radius of a circle so that it can be read by program `circle2.f90`. Run the program.

4. Alter program `circle4.f90` so that it prints a circle centred at the origin (0,0) with radius 4.0.

5. Define a derived type that could be used to store a date of birth in the following type of format:

   15 May 1990

   Write a program to test your derived type in a similar manner to the above examples. (`birth1.f90`)

6. Modify the derived type in exercise 5 to include a component for a name. (`birth2.f90`).

Control Structure:

7. Write a program containing a `DO` construct which reads numbers from the data file `square.dat`, skips negative numbers, adds the square root of positive numbers, and concludes if the present number is zero (use `EXIT` and `CYCLE`). (`sq_sum.f90`)

8. Write a program that reads in a month number (between 1 and 12) and a year number. Use the `CASE` construct to assign the number of days for that month, taking leap years into account. (`no_days.f90`)

9. Write a program that reads in a character string. Use the `CASE` construct in converting upper case characters to lower case and vice versa, and write out the new string. (Use `IACHAR("a") - IACHAR("A")` to determine the difference in the position in the collation sequence between lower and upper case characters.) (`convert.f90`)

Kind Values:

10. Run the program `kind_int.f90`. Notice how this program uses `SELECTED_INT_KIND` to find the kind values for integer variables on this system. Modify this program to find the kind values for real variables. (`kind_rl.f90`)

11. Run the program `mc_int.f90`. Notice how this program uses the kind values of integer variables found in exercise 1, and the numeric intrinsic functions to find some of the machine constants for this system. Modify this program by using the kind values of real variables found in exercise 1 and the numeric intrinsic functions (`PRECISION`, `HUGE`, `TINY` and `RANGE`) to find some of the machine constants for this system. (`mc_real.f90`)